Software Change in LA

Case study example

Thu, Sep 30, 2004

A software modification was needed recently for a local application. Making the change involved modifying both 68K and PowerPC versions of the LA. This note describes more precisely what steps were required to accomplish this typical job.

Background

In the protocol used for reporting Acnet alarm messages, each Acnet device is characterized by an error message code, or EMC. Originally, this was a unique 8-byte structure provided by the front end to identify the device that is the subject of the alarm message. The Acnet alarms handler, called `AEOLUS`, queried the Acnet database to translate the EMC into a device index, the main database key, which it then used to process the alarm. As the Acnet system evolved, a variation was allowed such that the front end could supply the device index, thus saving `AEOLUS` the translation step.

A note describing the details of how this feature was added to the IRM and Linac front ends is *Acnet Alarms via DI*, dated Apr 2, 2004. The key point here is that the new format of EMC includes the 4-byte device index and the Acnet node number. The other 2 bytes of the 8-byte structure are unused. But when this new version was put into service, `AEOLUS` preferred that this unused word be zero rather than the diagnostic value placed there. So the job at hand is to fix this by adding code that clears out the spare word in each EMC structure before sending the alarm message to `AEOLUS`.

The LA that handles delivery of Acnet alarm messages is called `AERS`. The source code is `AERS.p` for the 68K version used in IRMs. It is `LOOPAERS.c` for the PowerPC version. Both versions needed the modification, so that all nodes behave the same way.

Modification steps (68K)

To make the changes to the 68K version, the `AAUX.p` file was edited by the program editor `BBEdit` on the Macintosh G4, then compiled and linked by the MPW development system. To download it to library `node0508`, which houses the latest 68K versions of all the files, the following MPW tool command was used:

```
TFTPtool -w -n NODE0508 -r CODE -i 1 -s 4 LOOPAERS
```

This uses the `TFTP` protocol to send the executable file to the target node. The `TFTP` server in the target node, supported by local application `LOOPTFTP`, time stamps the new version, assigning the current time as a version date. When this file is transferred to other nodes, the version date is included and serves to identify the version in use.

The Download page application, `PAGEDNLD`, normally assigned as Page `D`, is used to transfer the file from the library node to any other node. This can be done by running this PA in any node. A new feature of transferring the file via multicast, however, requires that it be done from the source node, `node0508`. So one uses the “Page `G`’ mechanism to operate Page `D` on `node0508` to perform the file transfer, targeting the multicast node number `0x09F9`, which reaches all IRM and PowerPC nodes at once. But in this case, the PowerPC nodes will ignore this file transfer, because they see it as not being a PowerPC object program file. Also, during such a multicast transfer, only those nodes that already have a version of `LOOPAERS` will accept a new version. For more details on this file transfer support, see the recent note, *File System Update*, dated May 7, 2004. The end result is that in a single file transfer taking a second or so, all 120 IRMs are updated to the new version of `LOOPAERS`.

Modification steps (PowerPC)

To make the changes required for the PowerPC nodes, the file `LOOPAERS.c` was edited on `nova.fnal.gov` (or edited on the G4 and copied back to `nova`), and the following command was executed from the `vmeapps` directory:

```
make LOOPAERS
```

Then the following command was used to transfer the file, again using the `TFTP` protocol, to the library `node0619`, which houses all the PowerPC program files:

```
downloadapp node0619 LOOPAERS
```

In this case, `downloadapp` builds the version date from the file modification date and passes it along with the file to the `LOOPTFTP` server on the target node. (See the note, *File System Management*, dated Mar 23, 2000, for more on this.) Again, to share this version with the other PowerPC nodes, Page `G` was used to operate Page `D` on `node0619`, transferring the file to multicast node number `0x09FC`, which reaches all Linac front end nodes. (The target node number `0x09F9` would have worked just as well in this case.)

Finally, it is important to do a “`cvs commit`” for the new version of `LOOPAERS.c`.

Note that any target node accepting the new version of `LOOPAERS` automatically closes out the previous executing version and activates the new version. The result is that all nodes were very quickly running the modified version, without need for rebooting.